(51) **International Patent Classification⁷:** H04N

(21) **International Application Number:** PCT/US02/04989

(22) **International Filing Date:** 31 January 2002 (31.01.2002)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
60/265,519   31 January 2001 (31.01.2001)   US

(71) **Applicants and**
(72) **Inventors:** DODD, Timothy David [US/US]; 2126 Brownings Trace, Tucker, GA 30084-4628 (US). HEINRICH, Nicolas, [FR/FR]; 23 Avenue Ste. Marguerite, F-05200 Nice (FR).

(74) **Agent:** NEUFELD, Robert T.,; King & Spalding, 191 Peachtree Street, Atlanta, GA 30303-1763 (US).

(81) **Designated States** *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(54) **Title:** METHOD AND SYSTEM FOR CALCULATING RISK IN ASSOCIATION WITH A SECURITY AUDIT OF A COMPUTER NETWORK

ENVIRONMENT
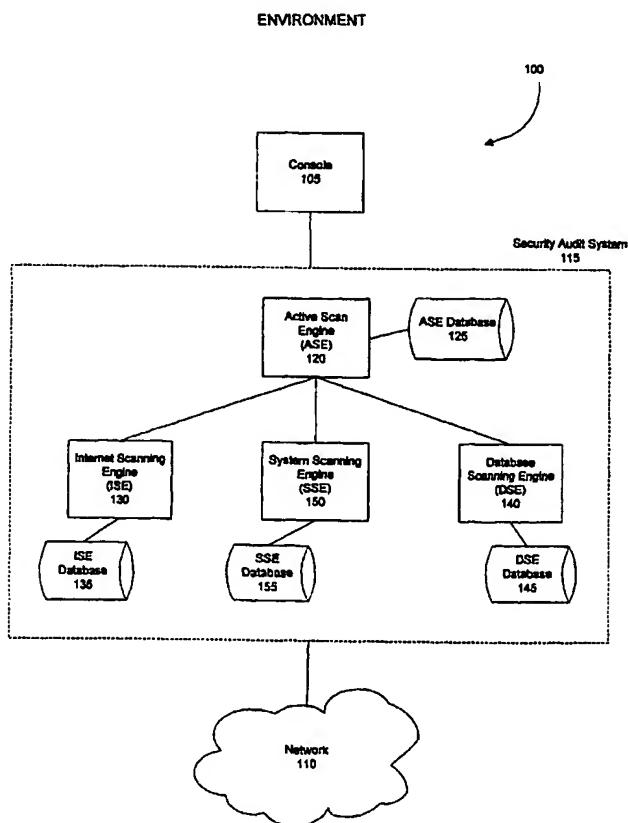
(57) **Abstract:** Calculating risk based on information collected during a security audit of a computing network. The computer network is surveyed to determine the significance of elements in the network and to identify vulnerabilities associated with the elements. Using this information, the security audit system calculates a risk value for each vulnerability. The risk value is a function of the asset value, the probability that the vulnerability will be exploited, and the potential severity of damage to the network if the vulnerability is exploited. The risk value can be adjusted based on the ease with which the vulnerability can be fixed. A network element may have one or more risk values associated with it based on one or more vulnerabilities. The security audit system employs a band calculation method for summing risk values and computing a single security score for the element. The band calculation method can also be used to produce a security score for a group of elements. The band calculation method produces a more accurate score for comparing elements and groups of elements throughout a network.

WO 02/062049 A2

METHOD AND SYSTEM FOR CALCULATING RISK IN ASSOCIATION WITH A SECURITY AUDIT OF A COMPUTER NETWORK

5   **PRIORITY AND RELATED APPLICATIONS**

The present application claims priority to provisional patent application entitled, "Method and System for Configuring and Scheduling Security Audits of a Computer Network," filed on January 31, 2001 and assigned U.S. Application Serial Number 60/265,519. The present application also references and incorporates herein a related U.S. non-provisional patent
10   application entitled, "Method and System for Configuring and Scheduling Security Audits of a Computer Network," filed concurrently herewith and having attorney docket number 05456.105009.

TECHNICAL FIELD

15   The present invention is generally directed to managing the security of a network. More specifically, the present invention facilitates computing a security score for elements in a distributed computing network.

BACKGROUND OF THE INVENTION

20   The security of computing networks is an increasingly important issue. With the growth of wide area networks (WANs), such as the Internet and the World Wide Web, people rely on computing networks to transfer and store an increasing amount of valuable information. This is also true of local area networks (LANs) used by companies, schools, organizations, and other enterprises. LANs typically are used by a bounded group of people in an organization to
25   communicate and store electronic documents and information. LANs generally are coupled to or provide access to other local or wide area networks. Greater use and availability of computing networks produces a corresponding increase in the size and complexity of computing networks.

With the growth of networks and the importance of information available on the networks, there is also a need for better and more intelligent security. One approach to securing
30   larger and more complex computer networks is to use a greater number and variety of security assessment devices. Security assessment devices can be used to evaluate elements in the network, such as desktop computers, servers, and routers, and to determine their respective

vulnerability to security problems, such as an attack from hackers. Security assessment devices can also be used more frequently to monitor the activity or status of the elements in a computing network. These network elements are commonly referred to as hosts. Throughout this specification the terms "host" and "element" will be used interchangeably to refer to the various

5   components that can be found in a distributed computing network.

However, simply increasing the number of security assessment devices and the frequency with which they are used does not solve the problems presented in conventional network security. With increased security activity, a network administrator or other user must decide which elements in the network need to be audited, how frequently they should be audited, and

10   what checks need to be run. These are decisions that often involve a variety of complicated factors and they are decisions that in practicality cannot be made every time a security audit is conducted. Increased assessment also produces a corresponding increase in the amount of security data that must be analyzed. A network administrator that is overwhelmed with security data is unable to make intelligent decisions about which security vulnerabilities should be

15   addressed first.

An additional difficulty associated with maintaining adequate network security is finding the time to conduct security audits. Security audits generally must be initiated by a security professional and can hinder or entirely interrupt network performance for several hours at a time. These limitations place a premium on the time available to conduct security auditing and

20   maintenance. Conventional network security systems do not support a means to accurately quantify security vulnerabilities so that they can be easily compared and prioritized.

In view of the foregoing, there is a need in the art for a system that will support the auditing of a distributed computing network. Specifically, a need exists to be able to automatically survey a network and prioritize any security issues identified by the survey. A

25   further need exists to be able to assess the security risk of each element in the network. The assessment should reflect the importance of the element and, for each security vulnerability that exists on the element, the ease with which the vulnerability can be exploited, and the impact of exploiting the vulnerability. Moreover, a need exists to accurately quantify the risk posed by vulnerabilities so that they can be compared in association with a particular host and so that hosts

30   can be compared over the entire network.

SUMMARY OF THE INVENTION

The present invention satisfies the above-described needs by providing a system and quantitative method for evaluating the security of elements in a network. A security audit system

5   can collect data concerning elements in a network. This data can include the operating system and services running on the element and any vulnerabilities associated therewith. This information can be used to calculate a risk for each vulnerability associated with an element. Certain elements may have few vulnerabilities and other elements may have many vulnerabilities. In order to give each element a meaningful security score, a banded calculation

10   method is used. The banded calculation method prevents many low-risk vulnerabilities associated with one element from overshadowing an element with a single high-risk vulnerability. This approach provides a simple means for a user to identify and address high-risk issues in a network.

In one aspect, the present invention comprises a method for computing a security score

15   associated with a host in a distributed computing network. A security audit system can select a vulnerability identified in a host and obtain an asset value for the host. The asset value is typically assigned to the host based on its characteristics and functions. The security audit system can also retrieve an exploit probability and a severity value for the vulnerability. Security personnel generally consider the various types of vulnerabilities and select predetermined exploit

20   probabilities and severity values. A risk value for a vulnerability can be computed from the host asset value, the exploit probability of the vulnerability, and the vulnerability's severity value. The risk value computation can be repeated for other vulnerabilities identified in the network. Because an element typically has multiple vulnerabilities, it is also useful to be able to compute a total security score for the element. The security audit system can use a banded calculation

25   model to compute the total security score by placing the risk values in selected bands on a risk scale. The banded calculation model prevents several low risk values from being summed and producing a disproportionately and inaccurately large security score.

In another aspect, the present invention provides a method for computing a risk value for quantifying a vulnerability identified in a network. A network security system can receive an

30   asset value for an element on which the vulnerability is detected. The asset value can be based on information collected during a security audit of the element. The network security system can

3

also receive a predetermined exploit probability and severity value for the vulnerability. Taking the asset value, the exploit probability, and the severity value, the network security system can compute a risk value that is useful in comparing other vulnerabilities in the network. The risk value can also be adjusted by a factor that reflects the difficulty of remedying the vulnerability.

5   These and other aspects of the invention will be described below in connection with the drawing set and the appended specification and claim set.

BRIEF DESCRIPTION OF THE DRAWINGS

       FIG. 1 is a block diagram illustrating an exemplary architecture for operating an
10 embodiment of the present invention.

       FIG. 2 is a logic flow diagram illustrating an overview of the operating steps performed by a security audit system in accordance with an exemplary embodiment of the present invention.

       FIG. 3 is a logic flow diagram illustrating an exemplary process for analyzing the results
15 of a security audit scan.

       FIG. 4 is a logic flow diagram illustrating an exemplary process for calculating an asset value for a host.

       FIG. 5 is a logic flow diagram illustrating an exemplary process for calculating a risk value for a vulnerability, and for accumulating these risk values for a single host.

20   FIG. 6 is a logic flow diagram illustrating an exemplary process for adjusting a vulnerability's calculated risk value as a function of the difficulty of fixing the vulnerability.

       FIG. 7 is logic flow diagram illustrating an exemplary process for assigning a vulnerability's risk value to a risk band, and for incrementing the count of vulnerabilities assigned to the band.

25   FIG. 8A is a diagram illustrating a representative example of a banded scale useful for calculating a security score from one or more risk values.

       FIG. 8B is a logic flow diagram illustrating an exemplary process for calculating a security score from the banded scale.

       FIG. 9 is a logic flow diagram illustrating an exemplary process for determining the
30 highest-risk band with at least one vulnerability risk value assigned to it.

FIG. 10 is a logic flow diagram illustrating an exemplary process for summing the risk values in each band on the banded scale.

DETAILED DESCRIPTION OF THE EXEMPLARY EMBODIMENTS

5   The present invention supports the assessment of the security risks of a computing network by providing a precise means to calculate and compare the risks posed by security vulnerabilities. Specifically, the present invention allows a security auditing system to identify security vulnerabilities in various elements throughout a network. The security auditing system also can collect information about the function and importance of elements in a computing
10   network. Using this information, the invention calculates a risk value for each security vulnerability that is identified. The risk value can be prioritized based on the ease with which the vulnerability can be repaired. Prioritizing risk values for a particular network element assists a user or network administrator in deciding which vulnerabilities to address first. The invention also supports the calculation of a security score for a network element that accumulates the risk
15   values of each vulnerability associated with the element. For example, employing a band calculation method ensures that a large number of low-risk vulnerabilities does not produce a higher security score than a smaller number of high-risk vulnerabilities. Calculating a security score for each element with the band calculation method allows for a more meaningful comparison of elements across a network.

20   Although the exemplary embodiments will be generally described in the context of software modules running in a distributed computing environment, those skilled in the art will recognize that the present invention also can be implemented in conjunction with other program modules for other types of computers. In a distributed computing environment, program modules may be physically located in different local and remote memory storage devices.
25   Execution of the program modules may occur locally in a stand-alone manner or remotely in a client/server manner. Examples of such distributed computing environments include local area networks of an office, enterprise-wide computer networks, and the global Internet.

The detailed description that follows is represented largely in terms of processes and symbolic representations of operations in a distributed computing environment by conventional
30   computer components, including database servers, application servers, mail servers, routers, security devices, firewalls, clients, workstations, memory storage devices, display devices, and

input devices. Each of these conventional distributed computing components is accessible via a communications network, such as a wide area network or local area network.

The processes and operations performed by the computer include the manipulation of signals by a client or server and the maintenance of these signals within data structures resident

5   in one or more of the local or remote memory storage devices. Such data structures impose a physical organization upon the collection of data stored within a memory storage device and represent specific electrical or magnetic elements. These symbolic representations are the means used by those skilled in the art of computer programming and computer construction to most effectively convey teachings and discoveries to others skilled in the art.

10   The present invention also includes a computer program that embodies the functions described herein and illustrated in the appended flow charts. However, it should be apparent that there could be many different ways of implementing the invention in computer programming, and the invention should not be construed as limited to any one set of computer program instructions. Further, a skilled programmer would be able to write such a computer program to

15   implement the disclosed invention based on the flow charts and associated description in the application text, for example. Therefore, disclosure of a particular set of program code instructions is not considered necessary for an adequate understanding of how to make and use the invention. The inventive functionality of the claimed computer program will be explained in more detail in the following description in conjunction with the remaining figures illustrating the

20   program flow.

Referring now to the drawings, in which like numerals represent like elements throughout the several figures, aspects of the present invention and the preferred operating environment will be described.

Fig. 1 illustrates various aspects of an exemplary computing environment in which an

25   embodiment of the present invention is designed to operate. Those skilled in the art will appreciate that Fig. 1 and the associated discussion are intended to provide a general description of representative computer network resources in an exemplary distributed computer environment including the inventive security audit system. The architecture comprises a console 105 and a security audit system 115 which are used to configure and schedule security audits of a network

30   110. The console 105 communicates information about the current security state of the network 110 to a user. The console 105 typically comprises a graphical user interface for presenting and

6

managing data in a convenient format for the user. The console 105 is also operable for
receiving information from the security audit system 115 and allowing control of the security
audit system 115. The security audit system 115 comprises an active scan engine 120 and one or
more other scan engines. In the exemplary embodiment illustrated in Fig. 1, the active scan

5   engine 120 is coupled to an Internet scanning engine 130, a system scanning engine 150, and a
database scanning engine 140. Each of these scan engines illustrated in Fig. 1 is coupled to a
corresponding database.

The active scan engine's 120 primary task is acquiring and maintaining current data
about the configuration and security posture of the network 110. The active scan engine 120

10   utilizes the subsidiary scan engines 130, 140 and 150 as a means for gathering information about
the network 110. The network 110 typically comprises elements such as desktop computers,
routers, and various servers. The active scan engine 120 is responsible for coordinating the
configuration, scheduling, and running of scans of these elements found in the network 110.
Typically, the active scan engine 120 is continuously running so that the scheduled scans can be

15   run at their designated times, and the resultant data processed in a timely manner.

Referring to Fig. 2, an overview of an exemplary security auditing process 200 is shown.
In alternative embodiments of the present invention, different auditing steps can be performed to
collect information about a network that is used to compute a security score. In step 205, an
active scan engine 120 configures scans that are to be run on a network 110. In step 210, the

20   active scan engine schedules the times at which the various configured scans will be run on the
network 110. An exemplary method for configuring and scheduling scans is described in greater
detail in the U.S. non-provisional patent application filed concurrently herewith and referenced
herein. When one of the scheduled times for running a scan is reached, the active scan engine
120 will run that scan on the network 110 in step 215. A scan may be run against particular

25   hosts in the network or over the entire network 110. Typically, a scan collects information about
the function of the hosts on the network and their respective vulnerabilities. This information is
gathered, analyzed, and used to compute a security score for each host or element in the network
in step 220. The security scores assist a user or network administrator in determining which
vulnerabilities should be addressed first.

30   Fig. 3 illustrates an exemplary process for analyzing the results of a security scan, as
referenced in step 220, for hosts located on the network 110. In step 305, the active scan engine

120 selects the ID of the first scan for which data are available. In step **310**, the active scan engine **120** selects the ID of the first host for which data are available in the current scan job results. In step **315**, the active scan engine **120** calculates an asset value for the current host. Asset values are calculated based on the operating system and services associated with the host.

5 An asset value is a characteristic representing the importance of a particular host or element in the operation of the network. In step **320**, the active scan engine computes a risk value for each vulnerability detected on the current host. The risk value quantifies the risk a particular vulnerability presents for a network. As will be discussed in connection with Figs. 7 and 8A, each risk value calculated for a vulnerability is assigned to a band on the risk scale. Using the

10 number of risk values assigned to each band, the active scan engine **120** computes a security score for the current host, in step **325**.

In step **330**, the active scan engine **120** determines if there are more hosts in the current job to be processed. If there are more hosts, the ID of the next host is retrieved in step **335**, and the process **220** returns to step **315**. In step **330**, if there are no more hosts to process in the

15 current job, the active scan engine **120** determines if there are more scan jobs to process in step **340**. If there are more scan jobs, the ID of the next scan job is retrieved in step **340**, and the process returns to step **310**. In step **340**, if the active scan engine **120** finds that there are no more hosts to process, the analysis is complete.

Fig. 4 illustrates an exemplary method for calculating an asset value as referred to in step

20 **315** of Fig. 3. Predetermined asset values are generally assigned for the various operating systems, host services, and the vulnerabilities that are likely to be encountered in a scan of a network. These predetermined asset values can be chosen by the provider of the security audit system or the user. The host's operating system and services are identified during the scan performed in step **215**. In step **405**, the asset value assigned to the host's particular operating

25 system is retrieved. In steps **410** and **415**, respective asset values are retrieved for the host's services and relevant vulnerabilities. In the exemplary calculation illustrated herein, the highest of these three asset values is selected in step **420** and used as the asset value for the host in computing the risk values in Fig. 3. In alternative embodiments of the present invention, other processes, such as averaging the asset values, may be used to designate the significance of a

30 particular host in a network **110**.

Fig. 5 illustrates an exemplary method for calculating risk values for the vulnerabilities associated with a host, as referred to in step 320 of Fig. 3. In step 505 the active scan engine 120 selects the first vulnerability discovered on the current host. In step 510, the active scan engine 120 retrieves the asset value for the current host calculated in Fig. 4. The active scan engine 120 retrieves a value representing the probability of a successful exploit against the current vulnerability in step 515. Each vulnerability has one or more exploit methods associated with it. The likelihood of a successful exploit is related to the difficulty of the exploit method. For example, certain vulnerabilities are easily exploited through the use of a simple, publicly available script. An easily exploited vulnerability is assigned a relatively high exploit probability value. On the other hand, other vulnerabilities are very difficult to exploit and involve a complex attack, such as the use of a buffer overflow. Because a buffer overflow is a difficult exploit method to implement, it is assigned a relatively low probability value. To retrieve a vulnerability's exploit probability value, the active scan engine 120 first retrieves the vulnerability's exploit method. The active scan engine 120 then consults a table that maps exploit methods to probability values. As outlined above, these values are assigned on the basis of the difficulty of each method. If a vulnerability has multiple exploit methods, the method with the highest exploit probability is selected.

In step 520, the active scan engine 120 retrieves a predetermined severity value for the current vulnerability. This value represents the impact of a successful exploit of the vulnerability. In step 525, the risk is calculated as the product of the host asset value and the vulnerability's severity and exploit probability values. Alternative embodiments of the present invention may weigh the three factors differently or use less than all three values in calculating the risk value. In step 530, the active scan engine 120 adjusts the calculated risk value to take into account the difficulty of fixing the current vulnerability. The user can choose not to incorporate the fix difficulty factor and instead, use the "pure" risk value. In step 530, the count of vulnerabilities assigned to the risk band that contains the adjusted risk value is incremented. In step 540, the active scan engine 120 retrieves the next vulnerability detected on the current host. If there is such a vulnerability, the process 320 returns to step 515. Otherwise, the calculation and accumulation of risk values for the host is complete.

Referring to Fig. 6, an exemplary process is illustrated for adjusting a risk value as referred to in step 530 of Fig. 5. In step 605, the active scan engine 120 retrieves a fix difficulty

value for the vulnerability. The fix difficulty value is a measure of the difficulty of remedying a vulnerability. Each vulnerability has associated with it one or more fix methods. To retrieve a fix difficulty value for a vulnerability, the active scan engine **120** first retrieves the vulnerability's fix method. The active scan engine **120** then consults a table that maps each fix

5   method to a difficulty value. This mapping table is typically created by network security personnel for use by the security audit system **115**. In step **610**, the risk value is multiplied by the fix difficulty value to yield an adjusted risk value. In step **615**, the adjusted risk value for the vulnerability can be compared to other adjusted risk values and ranked from highest to lowest. Using this exemplary method, when multiple vulnerabilities have the same computed risk, a user

10   of the security audit system can address those that are easier to fix first. The adjusted risk value can also be used to calculate the security score so that an element's security score will reflect the difficulty with which its vulnerabilities can be fixed.

Fig. 7 illustrates an exemplary method for assigning a risk value to a risk band, and for incrementing the count of vulnerabilities assigned to the band, as referred to in step **535** of Fig. 5.

15   A risk scale, as described in connection with Fig. 8, typically comprises several bands selected by the user. In step **705**, the active scan engine **120** identifies the band i, such that $r_i >= r(v) > r_{i-1}$, where r_i denotes the maximum risk value of band i, r_{i-1} denotes the maximum risk value of band $i-1$, and $r(v)$ denotes the risk of the current vulnerability. In step **710**, the count C_i of vulnerabilities assigned to band i is incremented by 1.

20   Fig. 8A and Fig. 8B illustrate an exemplary method for calculating a security score as referred to in Step **325** of Fig. 3. The illustrated exemplary method employs a logarithmic band calculation so that security scores can be accurately compared over the entire network **110**. Using the logarithmic band calculation prevents a host with a large number of low risk values from achieving a higher security score than a host with a few higher risk values. In the

25   illustration shown in Fig. 8A, risk values R_1, R_2, R_3, and R_4 are placed on a scale **800** that is divided into three bands. The number and width of the bands is arbitrary and can be varied to suit the needs and sensitivity of the network. The formula illustrated in Fig. 8A produces a security score that is a function of the band boundary values as opposed to the risk values. Using the exemplary risk values shown on the scale **800**, $r_{i(max)}$ is equal to 0.6, the upper boundary of

30   the highest band containing a risk value. In the exemplary formula,

$$\text{Security Score} = r_{i(max)-1} + \frac{r_{i(max)} - r_{i(max)-1}}{r_{i(max)} + r_{i(max)-1}} \sum_{k=1}^{|R|} \frac{r_{i(k)} + r_{i(k)-1}}{2^k}$$

the first term, $r_{i(max)-1}$, is 0.2, the lower boundary of the highest band containing a risk value. The remainder of the formula is a fraction of the middle band calculated from the band values encompassing the risk values placed on the scale **800**. The result of this calculation is that the security score will not exceed the upper boundary of the highest risk band, in this instance 0.6. The exemplary formula shown in Fig. 8A is merely one way of calculating a security score and alternative embodiments of the present invention may employ other formulas for the calculation.

Referring to Fig. 8B, an exemplary method for calculating a security score using the band calculation approach is illustrated. This exemplary method shows the calculation of a security score for a particular host in a network. This exemplary method can also be adapted to calculate a security score for a group of hosts within a network. Beginning with step **805**, the active scan engine **120** queries for the existence of vulnerabilities for the current host. If there are no vulnerabilities associated with the host, the security score is set to zero in step **810**. If there are vulnerabilities, the "yes" branch is followed to step **815**, where the active scan engine **120** determines the index of the highest-risk band that has at least one vulnerability's risk value assigned to it. As illustrated in Fig. 8A, the lowest band contains one risk value and the second band contains three risk values. In step **815**, the highest risk band which contains at least one risk value is identified as band $b_{i(max)}$. In the example shown in Fig. 8A, band $b_{i(max)}$ is the middle band bounded by 0.2 and 0.6 on the risk scale. As noted in Fig. 8A, an embodiment of the current invention implements a simplified version of the band calculation formula. This simplified calculation is accomplished in steps **820** and **825**. In step **820**, the simplified summation calculation yields preliminary score s. In step **825**, the security score is computed by multiplying the summation value by the fraction $(r_{i(max)} - r_{i(max)-1})/(r_{i(max)} + r_{i(max)-1})$, and adding $r_{i(max)-1}$, as noted in the formula above. In alternative embodiments of the present invention other methods may be used to sum the risk values to get an accumulated security score.

Fig. 9 illustrates an exemplary method for determining the highest-risk band in which a risk value is present. In the example shown in Fig. 8A, the second band is band $b_{i(max)}$ because there are no risk values in the upper region between 0.6 and 1.0. In step **905**, the index is set to the highest band risk. In step **910**, if the index is equal to b_{min}, the lowest risk band, the process

ends. Otherwise, in step **915** the active scan engine **120** determines if at least one risk value has been assigned to the current band.   If so, the current band index is the desired one and the process is complete.   Otherwise, the current index is decremented in step **920** and the process **815** returns to step **910.**

5   Fig. 10 illustrates an exemplary process for summing the banded vulnerability risk values as referenced in step **820.**   In step **1005**, the calculation is initialized by setting the current band index i equal to $b_{i(max)}$. The preliminary security score s is set to 0.0. The value *PreviousCount*, which represents the number of risk values that have already been processed, is set to 0. · The value *StartIndex*, which represents the index of the first risk value in the current band, is set to 1.

10   In step **1010**, if the current band is band b_{min}, the process is complete. Otherwise, control passes to step **1015**, where a value n is set equal to the count of risks C_i assigned to the current band plus the value *PreviousCount*. The preliminary score s is incremented by $(r_i + r_{i-1}) * (2^{n-StartIndex+1} - 1.0)/2^n$. The values *StartIndex* and *PreviousCount* are then incremented by C_i. The current band index i is then decremented by 1, and the process **820** returns to step

15   **1010.**

In conclusion, the present invention enables and supports security auditing of a distributed computing network by providing a useful numerical value of the risk associated with a vulnerability or group of vulnerabilities. The security audit system can collect information about the elements of a network and compute a risk value for vulnerabilities detected therein.

20   The risk value can be based on the importance of the network element, the likelihood of exploit, and the potential for damage to the network in the event the vulnerability is exploited. The risk value can also be adjusted to reflect the difficulty of remedying the vulnerability. The security audit system can also collect risk values for a particular element and compute a total security score for the element. The security audit system uses a banded calculation method to ensure that

25   a host with several low risk values does not have a higher security score than a host with a few high risk values. Security scores are useful for comparing individual elements or groups of elements on the network.

It will be appreciated that the present invention fulfills the needs of the prior art described herein and meets the above-stated objects.   While there has been shown and described the

30   preferred embodiment of the invention, it will be evident to those skilled in the art that various

modifications and changes may be made thereto without departing from the spirit and the scope of the invention as set forth in the appended claims and equivalence thereof. Although the present invention has been described as operating on a local area network, it should be understood that the invention can be applied to other types of distributed computing environments. Furthermore, it should be readily apparent that portions of the calculation can be varied in order modify the results without departing from the scope of the invention.

CLAIMS

What is claimed is:

5   1.   A method for assessing the security of a system comprising:

selecting a vulnerability for the system;

obtaining an asset value for the system;

determining an exploit probability for the vulnerability;

obtaining a severity value for the vulnerability;

10   computing a risk value for the vulnerability based on at least one of the asset value, the exploit probability, and the severity value;

if there are additional vulnerabilities associated with the system, repeating the foregoing steps to compute risk values for the additional vulnerabilities; and

calculating a security score for the system based on at least one of the risk values

15   associated with the system.

2.   The method of Claim 1, further comprising the step of calculating an adjusted risk value as a function of the risk value and a fix difficulty value.

20   3.   The method of Claim 2, further comprising the step of calculating an adjusted security score for the system based on at least one adjusted risk value.

4.   The method of Claim 1, further comprising the step of using the security score to assess the need for repair of the system.

25

5.   The method of Claim 1, further comprising calculating a group security score for a group of systems based on individual security scores for each of the systems.

6.   The method of Claim 1, wherein the asset value is obtained from at least one of an

30   operating system, a system service, and the system vulnerabilities.

7.   The method of Claim 1, wherein the severity value is based on the potential access available to the system from exploiting the vulnerability.

8.   The method of Claim 1, wherein the step of calculating a risk value comprises
5   multiplying the asset value, the probability of exploit value, and the severity value.

9.   The method of Claim 1, wherein the step of calculating a security score comprises placing a risk value on a banded scale.

10   10.   A computer-readable medium having computer-executable instructions for performing the steps recited in Claim 1.

11.   A method for computing a security score associated with a host in a distributed computing network comprising:

selecting a vulnerability for the host, the vulnerability being identified during a security scan;

5   obtaining an asset value for the host, the asset value obtained from at least one of a host operating system, a host service, and the host vulnerabilities;

determining an exploit probability for the vulnerability, the exploit probability indicating the likelihood that the vulnerability will be exploited to compromise the host;

obtaining a severity value for the vulnerability, the severity value characterizing

10   the potential damage that can be done from exploiting the vulnerability;

computing a risk value for the vulnerability based on at least one of the asset value, the exploit probability, and the severity value;

computing an adjusted risk value as a function of the risk value and a fix difficulty value, the fix difficulty value indicating the difficulty of remedying the vulnerability

15   associated with the risk;

if there are additional vulnerabilities associated with the system, repeating the foregoing steps to compute adjusted risk values for the additional vulnerabilities; and

calculating an adjusted security score for the host based on at least one of the adjusted risk values associated with the host.

20

12.   The method of Claim 11, further comprising the step of using the adjusted · security score to decide when to fix a host.

13.   The method of Claim 11, further comprising calculating a group adjusted security

25   score for a group of hosts based on individual adjusted security scores.

14.   The method of Claim 11, wherein the severity value is based on the potential access available to the network from exploiting the vulnerability.

30   15.   The method of Claim 11, wherein the step of computing a risk value comprises multiplying the asset value, the probability value, and the severity value.

16.   The method of Claim 11, wherein the step of calculating a security score comprises placing a risk value on a banded scale.

5   17.   A computer-implemented medium having computer-executable instructions for performing the steps recited in Claim 11.

18.   A method for determining a risk value for a vulnerability detected by a security audit system in a network comprising:

receiving an asset value from the security audit system for an element with which the vulnerability is associated;

receiving an exploit probability value for the vulnerability from the security audit system;

receiving a severity value from the security audit system; and

computing a risk value for the vulnerability, the computation comprising at least one of the asset value, the exploit probability value, and the severity value.

19.   The method of Claim 18, further comprising the step of computing a risk value for additional vulnerabilities associated with the element by repeating the foregoing steps.

20.   The method of Claim 18, further comprising the step of calculating a security score from at least one of the risk values associated with the element.

21.   The method of Claim 18, further comprising the step of computing an adjusted risk value as a function of the risk value and a fix difficulty value.

22.   The method of Claim 21, further comprising the step of calculating an adjusted security score from at least one adjusted risk value.

23.   The method of Claim 20, further comprising the step of calculating a group security score for a group of elements based on individual security scores.

24.   The method of Claim 18, wherein the asset value is based on at least one of a host operating system, a host service, and the host vulnerabilities.

25.   The method of Claim 18, wherein the severity value is based on the potential access available to the network from exploiting the vulnerability.

26.   The method of Claim 18, wherein the step of calculating a risk value comprises multiplying the asset value, the probability of exploit value, and the severity value.

5   27.   The method of Claim 18, wherein the step of calculating a security score comprises placing a risk value on a banded scale.

28.   A computer-readable medium having computer-executable instructions for performing the steps recited in Claim 18.

10

29.   A method for computing a risk value associated with an element in a network comprising:

receiving a vulnerability for the element, the vulnerability being identified by a

5   security audit system;

receiving an asset value for the element from the security audit system, wherein the asset value is based on at least one of an operating system, an element service, and the element vulnerabilities;

receiving an exploit probability value for the vulnerability from the security audit

10   system;

receiving a severity value from the security audit system; and

computing a risk value for the vulnerability, the computation comprising at least one of the asset value, the exploit probability value, and the severity value.

15   30.   The method of Claim 29, further comprising the step of computing a risk value for additional vulnerabilities associated with the element by repeating the foregoing steps.

31.   The method of Claim 30, further comprising the step of calculating a security score from at least one of the risk values associated with the element.

20

32.   The method of Claim 29, further comprising the step of computing an adjusted risk value as a function of the risk value and a fix difficulty value.

33.   The method of Claim 32, further comprising the step of calculating an adjusted

25   security score from at least one adjusted risk value.

34.   The method of Claim 31, further comprising the step of calculating a group security score for a group of elements based on individual security scores.

30   35.   The method of Claim 29, wherein the step of calculating a risk value comprises multiplying the asset value, the probability of exploit value, and the severity value.

36.   The method of Claim 29, wherein the step of calculating a security score comprises placing a risk value on a banded scale.

5   37.   A computer-readable medium having computer-executable instructions for performing the steps recited in Claim 29.

38.   A computer-readable medium having computer-executable instructions for performing steps comprising:

receiving a vulnerability for a host, the vulnerability being identified during a security scan;

obtaining an asset value for the host, the asset value based on at least one of a host operating system, a host service, and the host vulnerability;

determining an exploit probability for the vulnerability;

obtaining a severity value for the vulnerability;

computing a risk value for the vulnerability based on at least one of the asset value, the exploit probability, and the severity value;

computing an adjusted risk value as a function of the risk value and a fix difficulty value;

if there are additional vulnerabilities associated with the system, repeating the foregoing steps to compute adjusted risk values for the additional vulnerabilities; and

calculating an adjusted security score for the host based on at least one of the adjusted risk values associated with the host.

39.   The computer-readable medium of Claim 38, having further computer-executable instructions for performing the step of using the adjusted security score to decide when to fix a host.

40.   The computer-readable medium of Claim 38, having further computer-executable instructions for performing the step of calculating a group adjusted security score for a group of hosts based on individual adjusted security scores.

41.   The computer-readable medium of Claim 38, having further computer-executable instructions for performing the step of computing a risk value by multiplying the asset value, the probability value, and the severity value.

42.   The computer-readable medium of Claim 38, having further computer-executable instructions for performing the step of calculating a security score by placing a risk value on a banded scale.

43.   A system for computing a security score associated with a security audit of a distributed computing system comprising:

an manager software module operable for selecting a vulnerability for a host;

a storage module operable for storing an asset value for the host, an exploit

5   probability for the vulnerability, and a severity value for the vulnerability; and

a computation module operable for computing a risk value.

44.   The system of Claim 43, wherein the asset value for the host is based on at least one of the host's operating system, the host's services, and the host's vulnerabilities.

10

45.   The system of Claim 43, wherein computing the risk value is based on at least one of the asset value, the exploit probability, and the severity value.

46.   The system of Claim 43, wherein the computation module is further operable for

15   computing risk values for multiple vulnerabilities.

47.   The system of Claim 46, wherein the computation module is further operable for computing a security score from multiple vulnerabilities.

20   48.   The system of Claim 46, wherein the computation module computes a security score by placing multiple risk values on a banded risk scale.

49.   The system of Claim 47, wherein the computation module is further operable for computing a group security score from multiple security scores.

ENVIRONMENT

100

Console
105

Security Audit System
115

Active Scan
Engine
(ASE)
120

ASE Database
125

Internet Scanning
Engine
(ISE)
130

System Scanning
Engine
(SSE)
150

Database
Scanning Engine
(DSE)
140

ISE
Database
135

SSE
Database
155

DSE
Database
145

Network
110

FIG. 1

OVERVIEW OF
SECURITY
AUDITING

200

205

ASE Configures
Scans

210

ASE Schedules
Scans

215

ASE Runs Scans
of Network

220

ASE
Analyzes
Scan Results

FIG. 2

ANALYZE SCAN
RESULTS

FIG. 3

220

305
Get ID of First
Completed Scan
Job

310
Get ID of First
Host in Current
Scan Job

315
Calculate
Asset Value
of Current
Host

320
Quantify and
Accumulate
Vulnerability
Risks for the
Current Host

325
Compute
Security
Score of
Current Host

340
Get ID of Next
Completed Scan
Job

340
More Scan
Jobs?

330
Additional
Hosts in
Current Scan
Job?

335
Get ID of Next
Host in Current
Scan Job

yes

no

yes

no

Return

CALCULATE THE ASSET
VALUE OF A HOST

315

405

Retrieve the Asset
Value Assigned for
the Host's
Operating System

410

Retrieve the Asset
Value Assigned for
the Host's
Services

415

Retrieve the Asset
Value Assigned
any Relevant
Vulnerabilities

420

Store the Highest
of the Three Asset
Values as the
Host's Asset Value

FIG. 4

QUANTIFY AND ACCUMULATE
VULNERABILITY RISKS FOR A HOST

320

505

Select First Vulnerability v
Associated with the Host

510

Retrieve the Asset Value for
the Host

515

Retrieve an Exploit Probability
Value for Vulnerability v

520

Retrieve Severity Value for
Vulnerability v

525

Calculate Risk as Product of
Asset Value, Exploit
Probability, and Severity

530

Calculate Adjusted Risk
$r(v)$

535

Increment Vulnerability
Count C_i for the Risk Band
i that Contains $r(v)$

540

Get Next Vulnerability v
Associated with the Current
Host

545

Does v Exist?

yes

no

Return

FIG. 5

CALCULATE
ADJUSTED RISK FOR A
VULNERABILITY

530

605

Retrieve Fix Difficulty Value
for Vulnerability

610

Multiply the Calculated Risk by
the Fix Difficulty Value to Yield
Adjusted Risk

615

Rank Vulnerability According
to Adjusted Risk Value

FIG. 6

INCREMENT
VULNERABILITY COUNT
C_i FOR THE RISK BAND i
THAT CONTAINS RISK
$r(v)$

535

705

Identify the band i such that
$r_i >= r(v) > r_{i-1}$
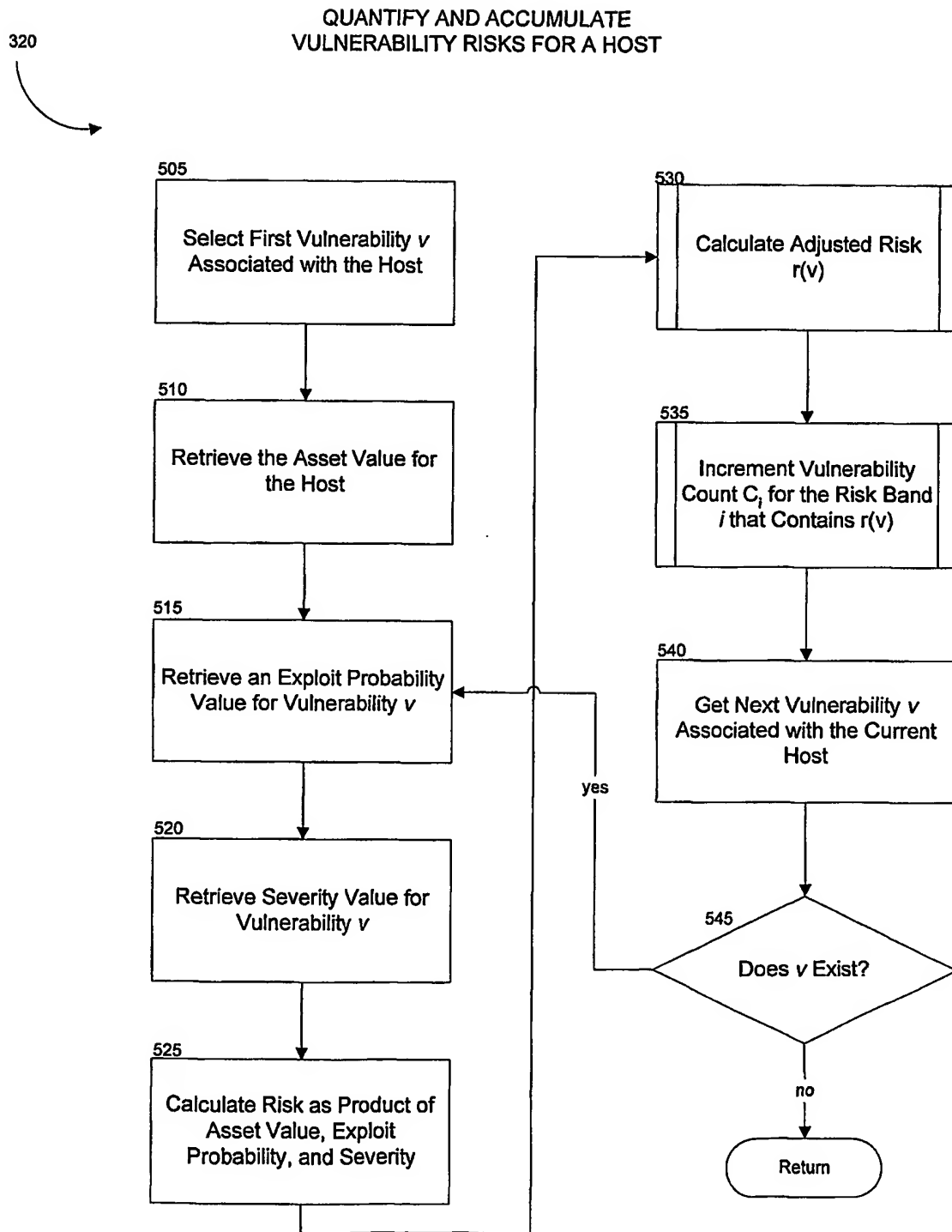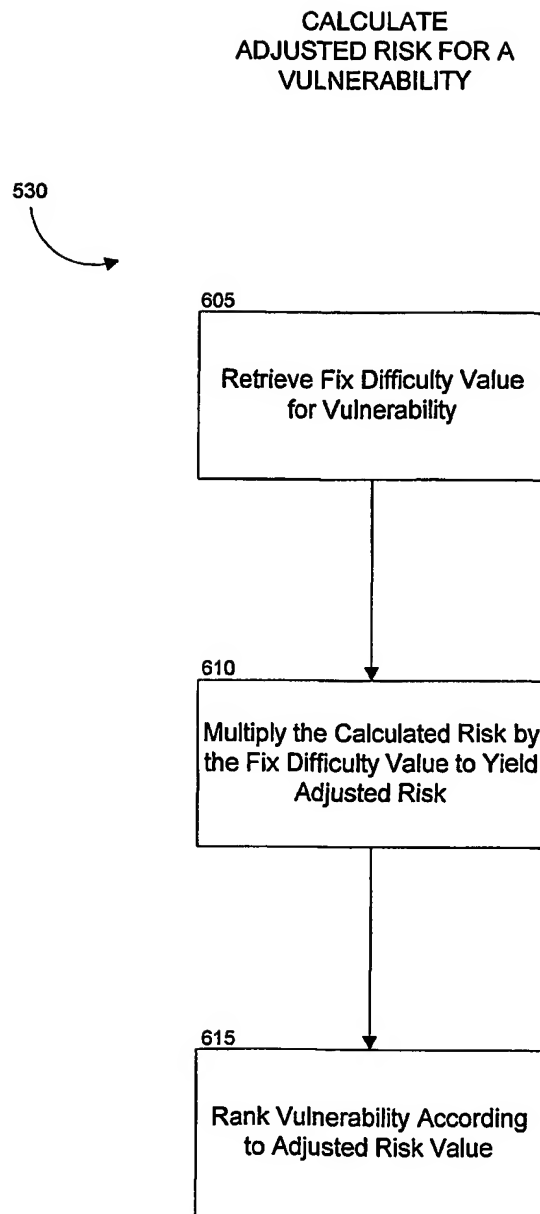
710

Increment the vulnerability
count C_i for band i by 1
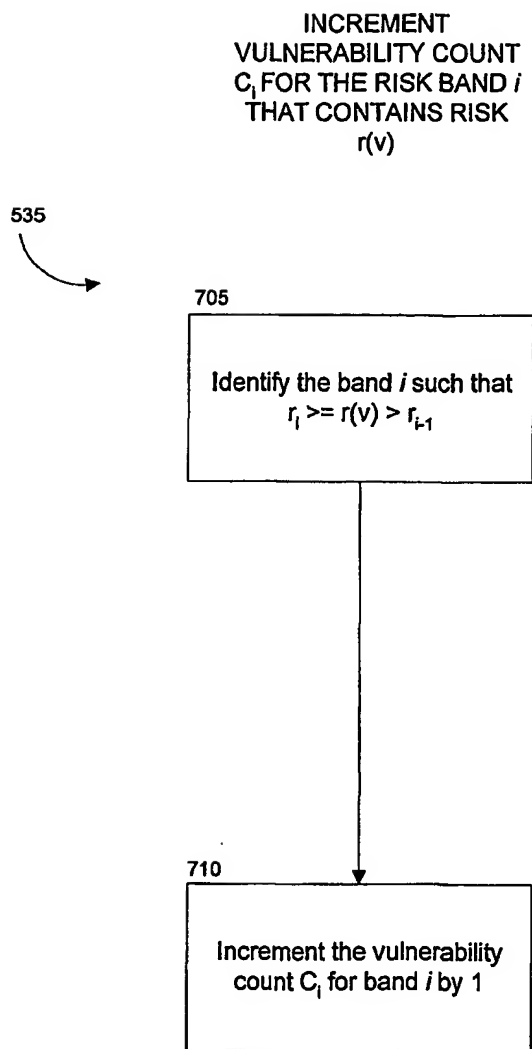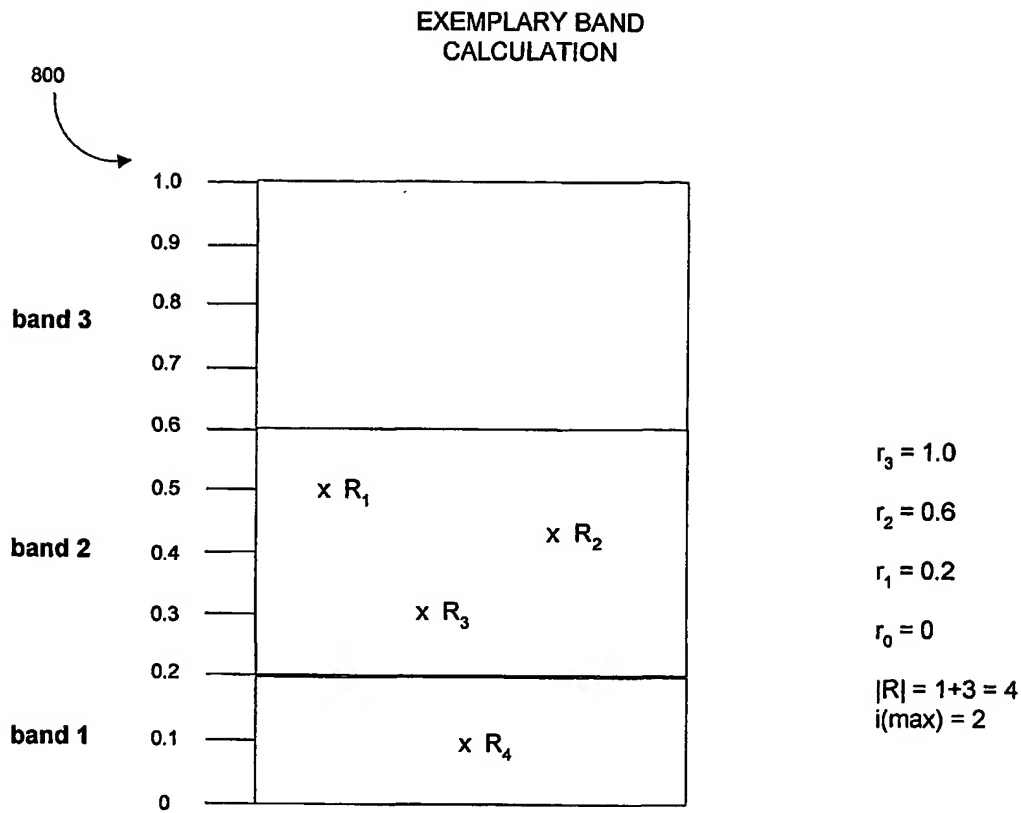
Notation:
- r_i = the maximum risk value of band i
- r_{i-1} = the maximum risk value of band $i - 1$
- $r(v)$ = the adjusted risk value of vulnerability v

FIG. 7

EXEMPLARY BAND
CALCULATION

800

$r_3 = 1.0$

$r_2 = 0.6$

$r_1 = 0.2$

$r_0 = 0$

$|R| = 1+3 = 4$
$i(\text{max}) = 2$

$$\text{Security Score} = r_{i(\text{max})-1} + \frac{r_{i(\text{max})} - r_{i(\text{max})-1}}{r_{i(\text{max})} + r_{i(\text{max})-1}} \sum_{k=1}^{\|R\|} \frac{r_{i(k)} + r_{i(k)-1}}{2^k}$$

Note:

Recognizing that the numerator ($r_{i(k)} + r_{i(k)-1}$) in the summation is constant for each band, and that

$$\sum_{k=i}^{n} \frac{1}{2^k} = \frac{2^{n-i+1} - 1}{2^n}$$

we can simplify the formula given above for computation purposes. The computations shown in subsequent figures implement the simplified method.

Notatation:

- i(max) = the index of the highest-risk band with at least one vulnerability
- r_i = the risk value at the upper boundary of band i
- i(k) = the index of the band that contains vulnerability k
- |R| denotes the number of vulnerabilities on the host

FIG. 8A

COMPUTE A HOST'S
SECURITY SCORE

325

805
Is the count of
vulnerabilities on the
current host > 0?

—no→

810
Set the security score = 0.0

yes

815
Find the highest-risk band
i for which $C_i > 0$; this is
band $b_{i(max)}$

820
Sum the bands, yielding
preliminary score s

825
Set $f_1 = r_{i(max)} - r_{i(max) - 1}$
Set $f_2 = r_{i(max)} + r_{i(max) - 1}$
Set $f_3 = (f_1/f_2)$
Security score = $f_3 * s + r_{i(max) - 1}$

FIG. 8B

FIND THE HIGHEST-
RISK BAND i FOR
WHICH $C_i > 0$; THIS IS
BAND $b_{i(max)}$

815

905

Set $i = b_{max}$

910

$i = b_{min}$? ——yes——> Return

no

915

$C_i > 0$? ————yes

no

920

Set $i = i - 1$

Notation:
- b_{max} = index of highest-risk band
- b_{min} = index of lowest-risk band

FIG. 9

SUM THE BANDS,
YIELDING
PRELIMINARY SCORE s

820

1005

Set $i = b_{i(max)}$
Set $s = 0.0$
Set PreviousCount = 0
Set StartIndex = 1

1010

$i >= b_{min}$? ——no——> Return

yes

1015

Set $n = C_i + $ PreviousCount
Set $s = s + (r_i + r_{i-1}) * (2^{n - StartIndex + 1} - 1.0)/2^n$
Set StartIndex = StartIndex + C_i
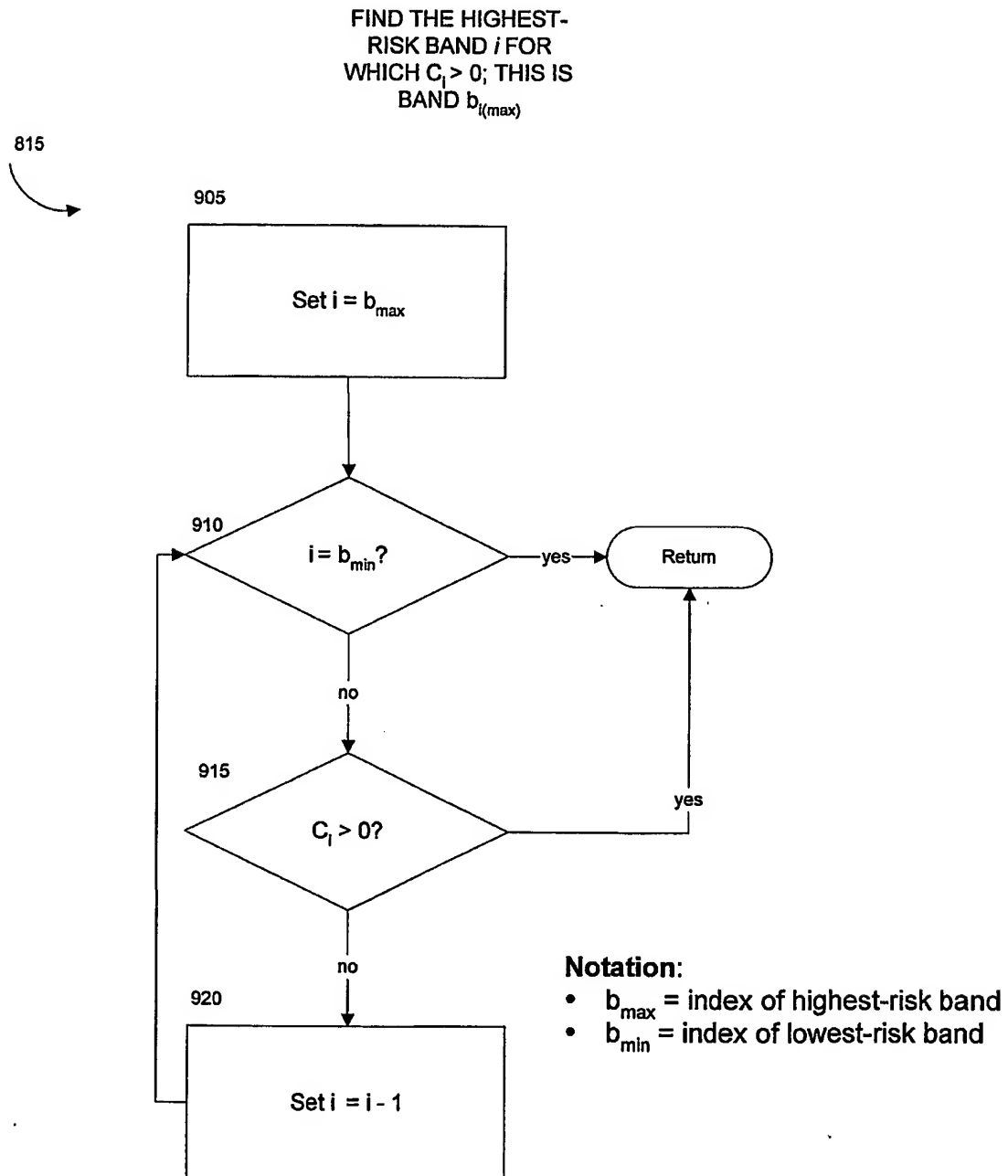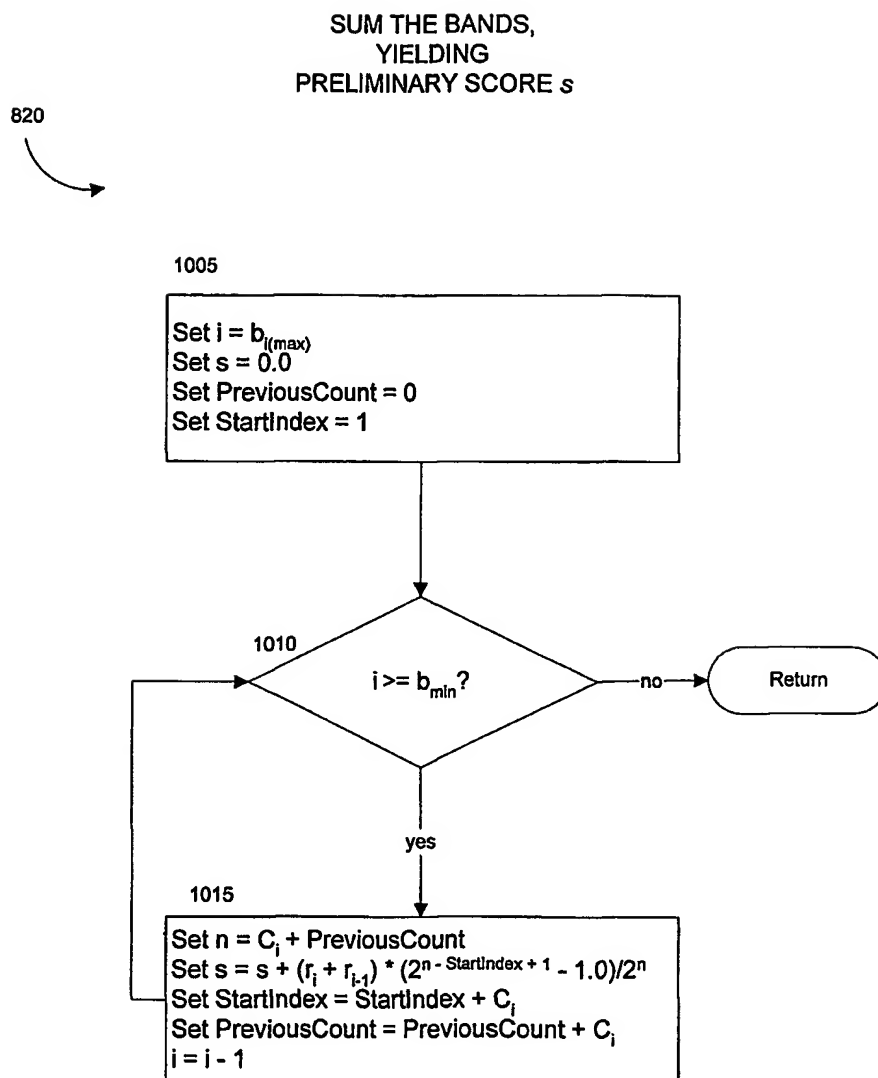Set PreviousCount = PreviousCount + C_i
$i = i - 1$

Notation:
- b_{min} = index of lowest-risk band
- $b_{i(max)}$ = index of highest-risk band with at least one vulnerability
- r_i = the maximum risk value of band i
- r_{i-1} = the maximum risk value of band $i - 1$

FIG. 10